RADIAL BASIS FUNCTIONS NETWORK FOR DEFECT SIZING

S. Nair, S. Udpa and L. Udpa
Center for Non-Destructive Evaluation
Scholl Road
Ames, IA 50011

INTRODUCTION

An important aspect of non-destructive testing is the interpretation and classification of signal obtained by NDT methods such as eddy current and ultrasound. These signals are typically complex, non-stationary waveforms, with signals corresponding to a particular class of defect in a specimen having similar form and shape. However, distortions and noise introduced by the measurement system make the manual classification of these signals a time-consuming and unreliable process, with the results affected by operator fatigue and measurement quality. The design of traditional classifiers for this task also poses many difficulties, due to a number of parameters that influence measurement, and the limited understanding of the effect of these parameters on the signal. Recently, artificial neural networks have been applied to a variety of NDT problems, including signal classification, with encouraging results. Artificial neural networks consist of a dense interconnection of simple computational elements, whose interconnection strengths are determined using a predefined learning algorithm, specific to the network. These networks do not require an explicit mathematical modeling of the data they have to process, and are robust even in the presence of noisy data and data generated by strongly non-linear processes [1]. An example of a neural network that has been extensively used in NDT applications is the multilayer perceptron. However, the error backpropagation algorithm used for training the multilayer perceptron has several disadvantages, such as long training times and susceptibility to local minima. This paper presents a novel approach to defect sizing that involves the use of a radial basis functions network. The network has the advantages of having shorter training times and a parametric nature that allows network optimization on an analytic basis. The application of such a network in the inversion of ultrasonic data to obtain flaw sizing is described. Results from the sizing of defects in aluminium blocks are presented.

THE MULTILAYER PERCEPTRON

Multilayer perceptrons are layered feed-forward networks, with one or more hidden layers of nodes between the input and output layers. Fig. 1 shows the architecture of a 3-layer multilayer perceptron, with an input, hidden and output layer. The input to the network is the n-dimensional vector to be classified. The nodes in the hidden and

Review of Progress in Quantitative Nondestructive Evaluation, Vol. 12
Edited by D.O. Thompson and D.E. Chimenti, Plenum Press, New York, 1993

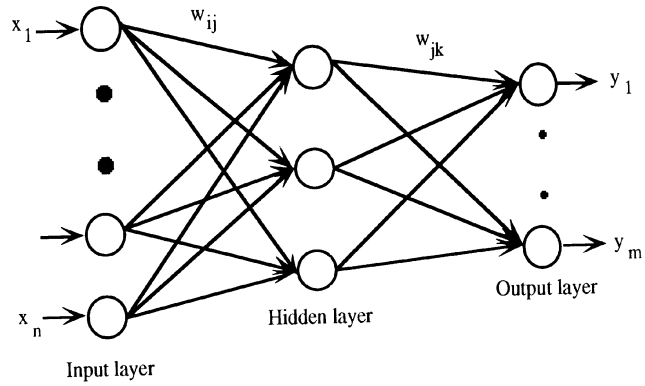

Figure 1. The multilayer perceptron.

output layers take as their input the weighted sum of the outputs of the nodes in the previous layer, and compute an output of the form

$$y_k = f\{ \sum_j w_{jk} x_j \} \qquad (1)$$

where x_k's are the node outputs of the previous layer, w_{jk}'s are the connection weights between the two layers, and f is the nonlinear transfer function of the node, usually chosen to be the sigmoid ($f(x) = 1/(1 + \exp(-\alpha x))$).

The input to the hidden layer nodes is, therefore, a weighted combination of the components of the data vector, and represents a hyperplane in the input pattern space. The output layer combines the hyperplane outputs of the hidden nodes, to form convex hulls that partition the pattern space into distinct regions. The input vector is classified on the basis of its location in the pattern space. The sigmoidal transfer functions of the hidden layer nodes serve to form curved hyperplanes, thereby bounding the convex hulls by smooth curves [2].

NETWORK TRAINING

Prior to classification, the multilayer percepton, as in the case of all neural networks, has to be trained. Training in learning networks is achieved by defining a cost function for the performance of the network in relating a set of known input-output pairs, called the training set, and minimizing this cost function by suitably adjusting the network parameters. The process of training serves to encode the higher order constraints of the input data into network parameters such as the weights and thresholds. In the pattern space, training can be visualized as the reorientation of the various hyperplane surfaces, which are initially in a state of random orientation, to positions where they form convex hulls capable of meaningful classification [3]. Following training, a network with N weights can be considered to represent a N-parameter family of models of the input data, which ideally is broad enough to adequately model the nonlinear input-output relationship [4]. This enables the network to have good generalization ability, so that the network is capable of classifying correctly input data not presented to it during training.

The multilayer perceptron uses the error backpropagation rule for training. In this algorithm, the weights and thresholds of the network are initialized to small random

values. Known input-output pairs from the training set are presented to the network, and the actual output of the network calculated. The least mean square error over the output nodes, E is calculated as

$$E = (1/2) \{ \sum_{j} (t_j - o_j)^2 \} \qquad (2)$$

where t_j is the known output, and o_j is the output calculated by the network. This error is then propagated backwards through the net, and the network weights adapted to minimize the error.

As seen from Equation (2), the error function used in training the multilayer perceptron is dependent on the response of nonlinear elements. The problem of minimizing this cost function is, therefore, one of unconstrained nonlinear least squares optimization, which necessitates the use of iterative techniques to find a globally optimum solution. The backpropagation algorithm uses a first order gradient descent technique to minimize the cost function. The algorithm, therefore, suffers from the drawbacks of gradient descent methods, such as slow convergence due to its tendency to oscillate in the proximity of the solution. Further, due to its susceptibility to local minima, the technique may converge to a spurious solution different from the true global minimum. Therefore, network training could be a lengthy process with no guarantee that the state of the network obtained at convergence is the optimum for the intended application.

The situation is further complicated by the fact that the relationship between network architecture and performance is poorly understood. Presently, the best network architecture for a particular application is decided on a heuristic basis, by experimenting with various topologies and choosing the one which offers the best performance. Since no criteria is available by which an optimum starting point can be chosen, the design process could entail a large number of trials to optimize the various parameters. The radial basis functions approach, in contrast, attempts to provide an analytic foundation to network design, by defining mathematically the implicit relationships existing between the network and the model it represents [4]. In doing so, the method also reduces drastically the time required to train a network by removing the need to run a number of trials before arriving at the final network configuration.

THE RADIAL BASIS FUNCTIONS APPROACH

The Radial Basis Functions (RBF) method is a technique in multivariable functional interpolation, which enables the implementation of a mapping from an n-dimensional input space into an m-dimensional output space in an RBF expansion as

$$s_k = \sum_{j=0}^{m} \lambda_{jk} \phi(||\mathbf{x} - \mathbf{c}_j||) \qquad (3)$$

where $\mathbf{x} \in \Re^n$, $\phi(\cdot)$ is the basis function chosen, $||\cdot||$ denotes some norm specified on $\Re^n$, and $\mathbf{c}_j$'s are the centres of the basis functions, which are fixed points in $\Re^n$. The coefficients λ_{jk} in the linear expansion determine the nature of the mapping that the function implements.

Since a trained neural network can also be considered as a mapping from one multidimensional space into another, the above method can be utilized in the design of such networks. The learning phase of the network would then be the optimization of a function which gives the best fit for the ordered input-output pairs, and which is constrained to go through all the known data points in the training set. The data points in the training set are assumed to be error free. The network can be designed to have generalization abilities by choosing a function that can interpolate along the constrained

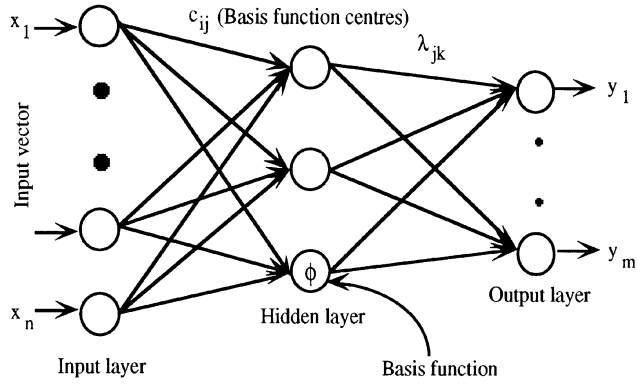

Figure 2. The radial basis network.

surface that is generated by the fitting procedure. The approximation and interpolation capabilities of the RBF expansion in Equation (3) makes it a good choice as the mapping function. Further, this function has the form of a weighted sum over nonlinear functions, which implies that this can be implemented in a network with a topology similar to that of the multilayer perceptron. Such an implementation is shown in Fig. 2. In this network, the input to the network is again the data vector $\mathbf{x}$, the connections between the input and hidden layers are the radial basis function centres $\mathbf{c}_j$, and the fan-in to the hidden nodes is the norm $\| \bullet \|$. The transfer function of the hidden node is chosen as the basis function, so that the hidden nodes compute $\phi (\| \mathbf{x} - \mathbf{c}_j \|)$. The coefficients λ_{jk} are placed between the hidden and output nodes, which have a transfer function of unity, so that the network computes the mapping function shown in Equation 3.

TRAINING THE RBF NETWORK

The parameters in the radial basis functions network that can be varied during training for optimizing network performance are the basis functions ϕ, the centres for the basis functions $\mathbf{c}_j$, and the feedforward coefficients λ_{jk}. A typical choice for the basis function is the gaussian ($\phi(x) = \exp[-x^2/\sigma]$). Functions with good interpolation properties, such as splines can be used as basis functions [5]. The centres for the basis function must be chosen to sample the input space appropriately and may be chosen as a suitable subset of the training data or distributed uniformly in the data space [6]. Once the basis functions and their centres have been decided, the known input-output values in the training set can be substituted into Equation 3, to reduce it to a set of linear equations, which can be solved to obtain the feedforward coefficients.

The training process for the network, therefore, is essentially the determination of the coefficients λ_{jk}, which can be obtained by the solution of a set of linear equations. This is in contrast to the multilayer perceptron, which uses unconstrained nonlinear least squares optimization to determine its network coefficients. Therefore, the radial basis network is not susceptible to local minima, and is guaranteed to yield a solution. Further, since no iterative technique is needed to obtain the solution, the network has a significantly shorter training time than the multilayer perceptron.

RESULTS

A radial basis network was implemented to size defects in aluminium blocks based on their ultrasonic signals. In order to compare their performances, a multilayer perceptron was also trained and tested on the same signals. A total of 270 ultrasonic

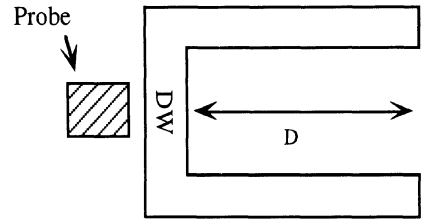

Figure 3. ASTM reference block with defect.

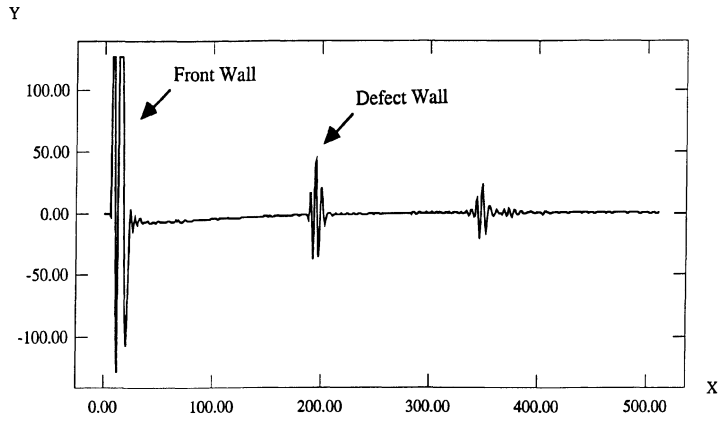

Figure 4. Typical defect signal

signals were obtained from a set of defects of depths 0.12, 0.25, 0.38, 0.50, 0.62, 0.75, 0.88 and 1.00 inches, machined into ASTM reference blocks. The aluminium block with defect and the position of the probe are as shown in Figure 3. A typical defect signal is shown in Figure 4, exhibiting a strong front wall reflection, followed by a weaker reflection from the defect. Increasing defect depth is indicated by decreasing time of flight between front wall and defect signal.

Prior to the presentation of signals to the networks for processing, the strong front wall reflection was removed from all signals, so as to make the networks more sensitive to the defect reflection. Two simple preprocessing schemes to reduce data dimensionality, namely coarse coding [7] and decimation, were tried. In the first method, the signal was split into a set of bins, and each bin replaced by its average. In the second method, the signal was sampled equally, with sufficient samples to preserve information. It was found that decimation yielded better results, and was chosen as the preprocessing scheme in the results presented.

Two properties of the networks, generalization and interpolation, were compared during testing. In the first test, the ability of the networks to size the given defects accurately was evaluated. For this, one-third of the signals from all the nine defect classes were used in training the network, and the networks were then tested on the remaining two-thirds of the data. Outputs were considered to be correct if the error in defect depth calculated by the network was less than ten percent of the actual depth. The second property tested was the ability of the networks to size signals from defects with depths different from those presented during training. For this, signals from three defect classes (depths 0.38, 0.50 and 0.62 inches) were suppressed during training, and the networks were trained on one-thirds of the signals from the remaining six defect classes. Signals not presented during training were used for testing. A similar error criterion as in the first case was applied to decide correct classification.

After experimenting with various architectures, a network configuration of 64 input, 32 hidden and one output node was chosen for the multilayer perceptron, as it gave the best results with respect to both sizing and interpolation. For the radial basis network, the basis function centres were chosen as a subset of the training data set, by applying the maximin clustering algorithm [3] on the training data set and using the cluster centers so determined as the basis function centres. The RBF networks obtained for the two training sets had 38 and 35 centers, with 64 input nodes and one output node in both cases. The basis function chosen was the logarithmic function ($\phi(x) = \log(1 + x)$).

The multilayer perceptron with backpropagation algorithm required a training time of approximately four hours. In contrast, the radial basis network, with the matrix pseudo-inversion to obtain the feedforward coefficients done using singular value decomposition, required a training time of approximately two minutes. This represents a significant drop in training time. Both networks achieved a classification performance of 100% on the training data in both the tests.

The performance results of the two networks on testing data are shown in the graphs in Figures 5 and 6. These illustrate plots of the actual classification of the networks versus the true (or desired) classification. Ideally the outputs of the network would have fallen on a line of unit slope passing through the origin. The divergence of the calculated outputs from this line indicates error made by the networks. It is seen that the error in the case of the multilayer perceptron is much greater than that of the RBF network. Using the criterion described earlier, the classification performances of the two networks are as shown in Table 1. The performance of the radial basis network is better than the multilayer perceptron in both the cases. The superiority of the RBF network is seen in the interpolation test, where it is able to classify correctly almost all the signals not presented to it during training. In contrast the multilayer perceptron does a poor job. This demonstrates the ability of the RBF network to interpolate in the high dimensional input-output space.

CONCLUSIONS

The radial basis functions network is a viable alternative to the multilayer perceptron for NDE applications. The learning method used by the network offers a guaranteed solution, with faster training times. Moreover, instead of using a trial and error approach in network design, any apriori information about the nature of the data being processed can be employed to determine the basis functions to be used and the location of the centres in the data space. Once these parameters are chosen, the training process reduces to the solution of a set of linear equations. This eliminates the need for heuristic techniques to optimize network parameters. This network is particularly suited for applications which require a continuous valued output, such as defect sizing and continuous valued input-output mapping, and is superior to the multilayer perceptron in such applications. The ability of the network to interpolate in high dimensional spaces enables it to recognize classes not presented to it during training.

Table 1. Classification results of the multilayer perceptron and radial basis network.

	Training time	% Classification	
		Generalization	Interpolation
Multilayer Perceptron	4 hours	74.3 %	61.0 %
RBF Network	2 min.	86.0 %	84.1 %

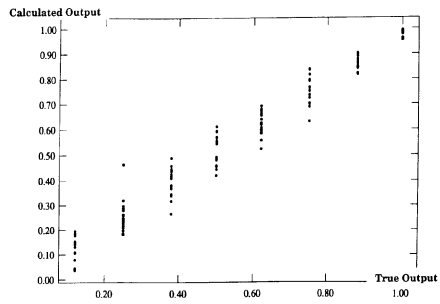

Fig 5a. Multilayer Perceptron: Generalization.

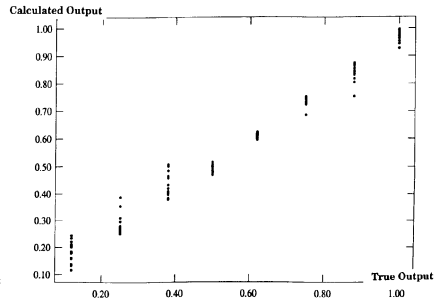

Fig 5b. RBF Network: Generalization.

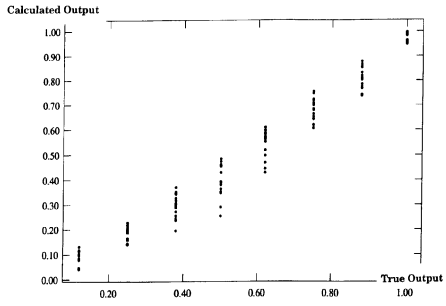

Fig 6a. Multilayer Perceptron: Interpolation.

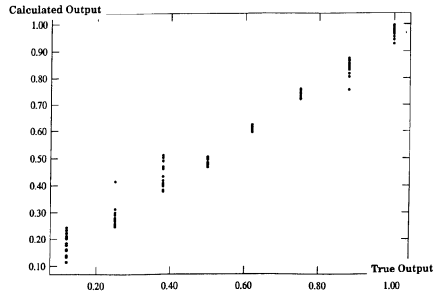

Fig 6b. RBF Network: Interpolation.

ACKNOWLEDGEMENTS

This work was sponsored by the FAA-Center for Aviations Systems Reliability, operated by the Ames Laboratory, USDOE, for the Federal Aviation Administration under Contract No. W-7405-ENG-82 with Iowa State University.

REFERENCES

1. R. P. Lippman, IEEE ASSP Magazine, Vol. 4, p. 4 (1987).
2. R. Beale and T. Jackson, *Neural Computing - An Introduction*, IOP Publishing, Britain (1988).
3. J. T. Tou and R. C. Gonzalez, *Pattern Recognition Principles* , (Addison-Wesley, 1974).
4. D. S. Broomhead and D. Lowe, Complex Systems, Vol. 2, p. 321-355 (1988).
5. Yiu-Fai Wong, IEEE Transactions on Neural Networks, II - 133 (1991).
6. S. Chen, A. Billings and W.Luo, Int. J. Control, Vol. 50, No. 5, p. 1873 (1989).
7. D.E. Rummelhart and J. L. McClelland eds., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol. 1, MIT Press (1988).